

**Remarks**

The applicants have carefully reviewed the Office action dated June 22, 2007. The rejections of the claims are respectfully traversed. For at least the reasons provided below, it is respectfully submitted that all claims are in condition for allowance. Withdrawal of the rejections to the claims and allowance thereof are respectfully requested.

As an initial matter, claims 1, 7, 14, and 19 have been amended to include the recitations of dependent claims 5, 11, 18, and 23 and claims 5, 11, 18, and 23 have been cancelled to place the application in better form for appeal by simplifying the issues to be appealed. Accordingly, the applicants respectfully request that the amendments be admitted to the record.

Claim 1 recites a method comprising, *inter alia*, rejecting a driver request, wherein rejecting the driver request comprises storing the protocol interface in a data structure in response to identifying a request by a driver to access an architectural protocol installed in the processor system.

The Office action rejected claims 1 and 5, among other claims, as unpatentable under 35 U.S.C. § 103 over “Extensible Firmware Interface Specification” version 1.02 (“Version102”) in view of Blumenau et al. (US 6,993,581) (“Blumenau”).

The applicants respectfully submit that the rejection of claim 5, rejecting the recitations now present in claim 1, is in error and must be withdrawn. The portions of Blumenau cited in the Office action do not describe or suggest that rejecting the driver request comprises storing the protocol interface in a data structure in response to identifying a request by a driver to access an architectural protocol installed in the processor system. Rather, Blumenau states, “the security driver 42 denies the request, signals an exception back to the maker of the request, and terminates processing of the command.” (Col. 10, lines 19-21). In other words, while Blumenau describes several procedures that are performed as part of rejecting a request, Blumenau does not suggest that rejecting the driver request comprises storing the protocol interface in a data structure in response to identifying a request by a driver to access an architectural protocol installed in the processor system. Even if the security driver of Blumenau rejecting a request is the same as rejecting a driver request as

recited in claim 1, a point that the applicants do not concede, Blumenau fails to describe the noted recitation of claim 1.

The Office action seeks to cure the deficiencies of Blumenau by citing Version102. The Office action admits that Version102 does not teach rejecting a driver request, wherein rejecting the driver request comprises storing the protocol interface in a data structure in response to identifying a request by a driver to access an architectural protocol installed in the processor system. (Office action, 3:19). However, the Office action contends that Version102 describes “a protocol interface that was installed and reinstalled. (Office action, 3:20-21). Further, the Office action contends that it would have been obvious to have recognized that installed protocol interfaces are stored within storage media in a processor system. (Office action, 3:21-4:1). However, the applicants respectfully submit that even if the examiner’s contention that storage of protocol interfaces in storage media is correct, a point which the applicants do not concede, claim 1 recites (as was recited in claim 5) rejecting a driver request, **wherein rejecting the driver request** comprises storing the protocol interface in a data structure **in response to identifying a request by a driver to access an architectural protocol installed in the processor system.** The Office action merely alleges that Version102 describes storing a protocol interface in data structure. The Office action does not suggest that Version102 describes that storing the protocol interface in a data structure is performed in rejecting a driver request nor does the Office action suggest that storing the protocol interface in a data structure is performed in response to identifying a request by a driver to access an architectural protocol installed in a processor system.

It is well established that the prior art must teach or suggest each of the claim elements ... to establish a *prima facie* case of obviousness. See *In re Oetiker*, 24 USPQ. 2d 1443, 1446 (Fed. Cir. 1992); *Ex parte Clapp*, 227 USPQ. 972, 973 (Bd. Pat. App. 1985); *In re Royka*, 490 F.2d 981 (CCPA 1974) and M.P.E.P. § 2143. Neither of Blumenau nor Version102 describes or suggests rejecting a driver request, wherein rejecting the driver request comprises storing the protocol interface in a data structure in response to identifying a request by a driver to access an architectural protocol installed in the processor system. In fact, as described above, the Office action fails to even allege that either of Blumenau, Version102, or a combination thereof describes or suggests rejecting a driver request, wherein rejecting the driver request comprises storing the protocol interface in a data

structure in response to identifying a request by a driver to access an architectural protocol installed in the processor system. Therefore, for at least the forgoing reasons, claim 1 and all claims depending therefrom are in condition for allowance.

Claims 7, 14, and 19 respectively recite a machine readable medium instructions storing instructions, which when executed, cause a machine to, *inter alia*, reject a driver request by storing the protocol interface in a data structure in response to identifying a request by a driver to access an architectural protocol installed in the processor system, an apparatus comprising, *inter alia*, a processor operatively coupled to the data structure and to a machine readable medium storing instructions that, when executed, cause the processor to, *inter alia*, reject the driver request, and to store the protocol interface in the data structure in response to identifying a request by a driver to access an architectural protocol installed in the processor system, and a processor system to protect a protocol interface comprising, *inter alia*, a processor operatively coupled to the DRAM and to a machine readable medium storing instructions that, when executed, cause the processor to, *inter alia*, reject the driver request, and store the protocol interface in the data structure in response to identifying a request by a driver to access an architectural protocol installed in the processor system. As described in conjunction with claim 1, none of Blumenau, Version 102, or any combination thereof describes or suggests rejecting a driver request, wherein rejecting the driver request comprises storing the protocol interface in a data structure in response to identifying a request by a driver to access an architectural protocol installed in the processor system. Therefore, for at least the forgoing reasons, claims 7, 14, 19, and any claims depending therefrom are in condition for allowance.

**U.S. Serial No. 10/608,586**  
**Response to the Office action of June 22, 2007**

If there is any matter that the examiner would like to discuss, the examiner is invited to contact the undersigned representative at the telephone number set forth below.

Respectfully submitted,  
HANLEY, FLIGHT & ZIMMERMAN, LLC  
150 South Wacker Drive  
Suite 2100  
Chicago, Illinois 60606

Dated: August 22, 2007

/Michael W. Zimmerman/  
Michael W. Zimmerman  
Reg. No. 57,993  
Agent for Applicants  
312.580.1020